



# Blockchain Building Blocks

FinTech  
Lesson 18.3



# Class Objectives

---

By the end of this unit, you will be able to:

01

Explain the most popular consensus algorithms and the tradeoffs between each.

02

Create a genesis block using `puppeth`.

03

Initialize `geth` nodes using a `genesis.json`.

04

Run and connect `geth` nodes together.

05

Build a blockchain network and produce blocks.

06

Send a transaction on your local network.

# Blockchain Skill Check

---

Let's refresh a bit on the data structure of a blockchain.

**What does the “chain” in blockchain refer to?**



The chain of hashes that link each block to the previous.

**What is a digital signature?**



A message that you can validate the integrity of and authenticity of cryptographically.

**What is a node?**



A participant in the network that maintains a full copy of the blockchain.

# Building a Blockchain

# Building a Blockchain

The background of the slide is a dark blue gradient. It features several glowing, translucent cubes that appear to be floating. Some of these cubes have binary code (0s and 1s) on their faces. A network of thin, white lines connects various points, resembling a blockchain or a neural network. The overall aesthetic is futuristic and technological.

What we are going to do today?

01

Build a blockchain from scratch!

02

Learn the differences between the various consensus algorithms available.

03

Make transactions in our very own blockchain.

# Consensus Algorithms

# Consensus Algorithms

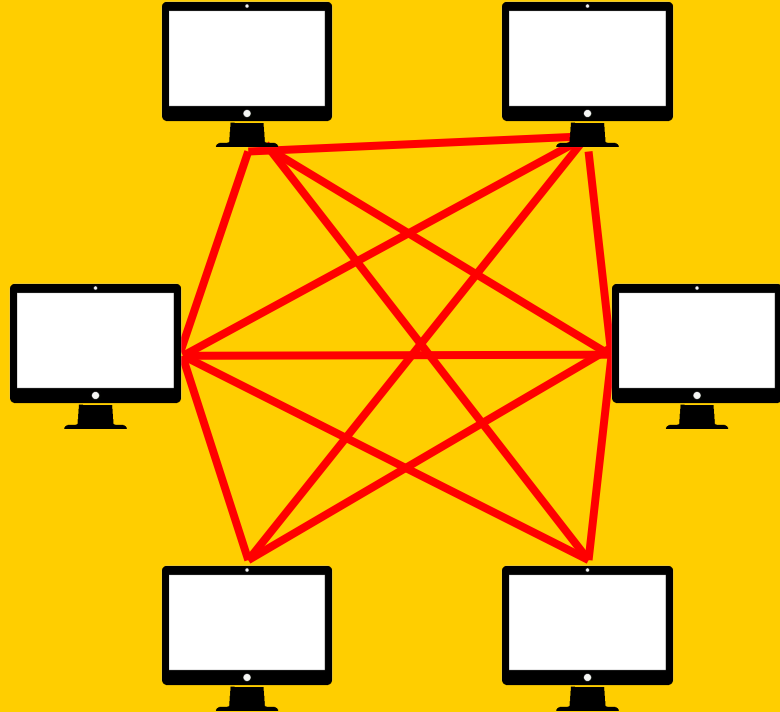
---

In a decentralized system, you cannot trust the participants in the network.

It's a database that can be written to by anyone, which means special rules must be in place to prevent the database from being modified in a malicious way. This is where something called a **"Consensus Algorithm"** comes into play.

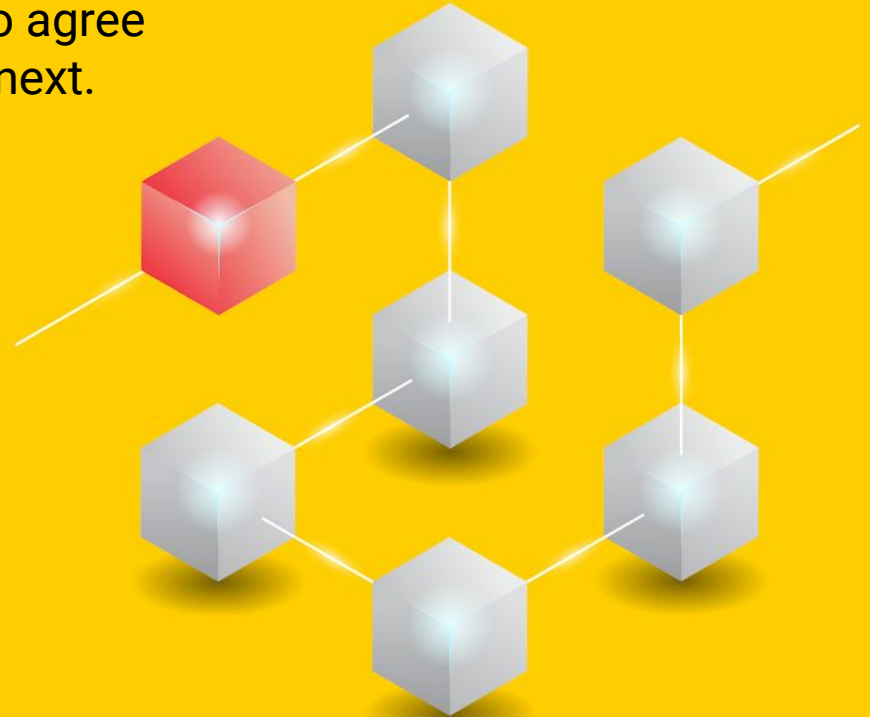
---

## Decentralized Database



# Consensus Algorithms

The main purpose of a consensus algorithm in blockchain is to get the entire network to agree on which block gets added to the chain next.





# Consensus Algorithms

---

Let's discuss the three most popular algorithms relevant to the blockchain.

01

Proof of Authority  
(PoA)

02

Proof of Work  
(PoW)

03

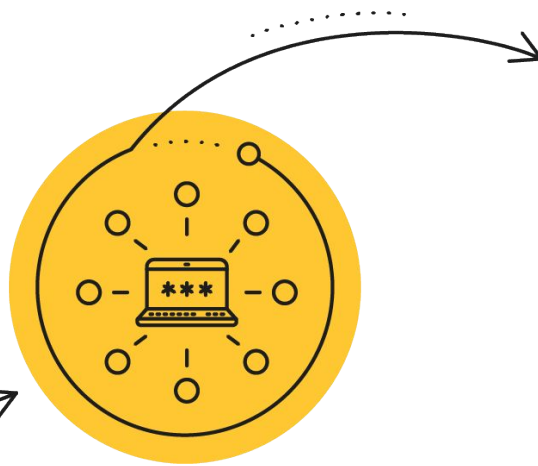
Proof of Stake  
(PoS)

# Consensus Algorithms

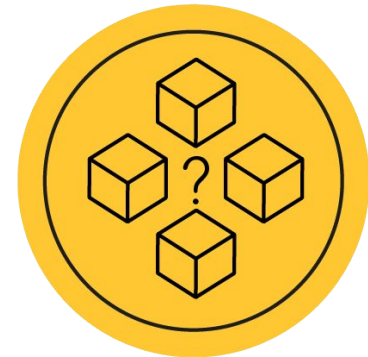
## Proof of Authority (PoA)



Allows only specific addresses to mine/produce blocks in the network



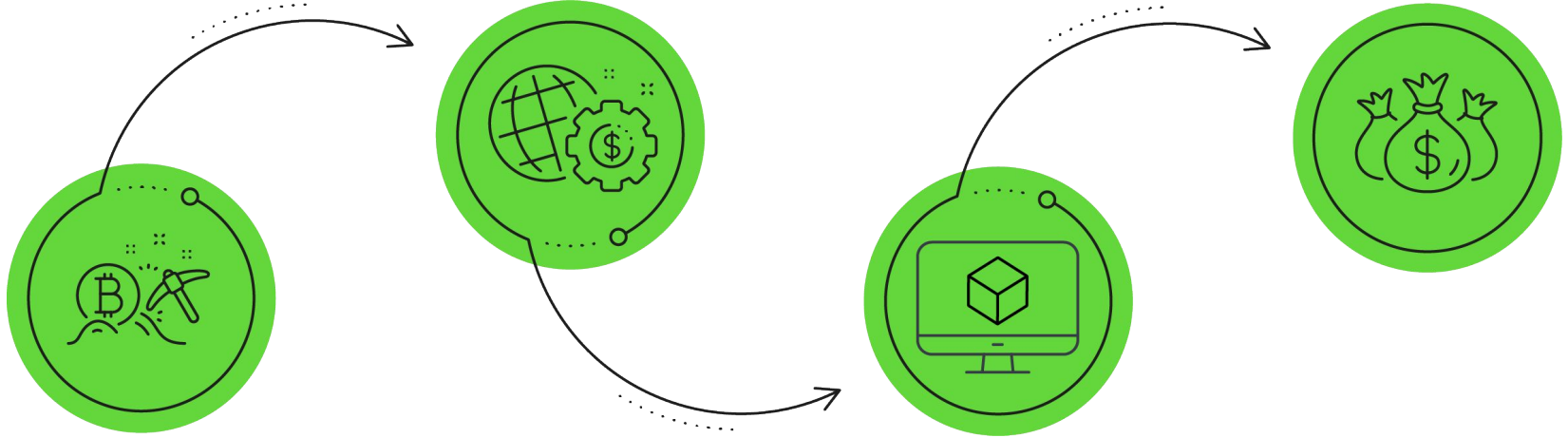
A centralized but cheap algorithm mainly used to power test networks



Never used in production of mainnet blockchains, only for development and testing in testnet blockchains

# Consensus Algorithms

## Proof of Work (PoW)



The most popular algorithm in blockchain currently. This is what Bitcoin came out with, and where the term “mining” comes from.

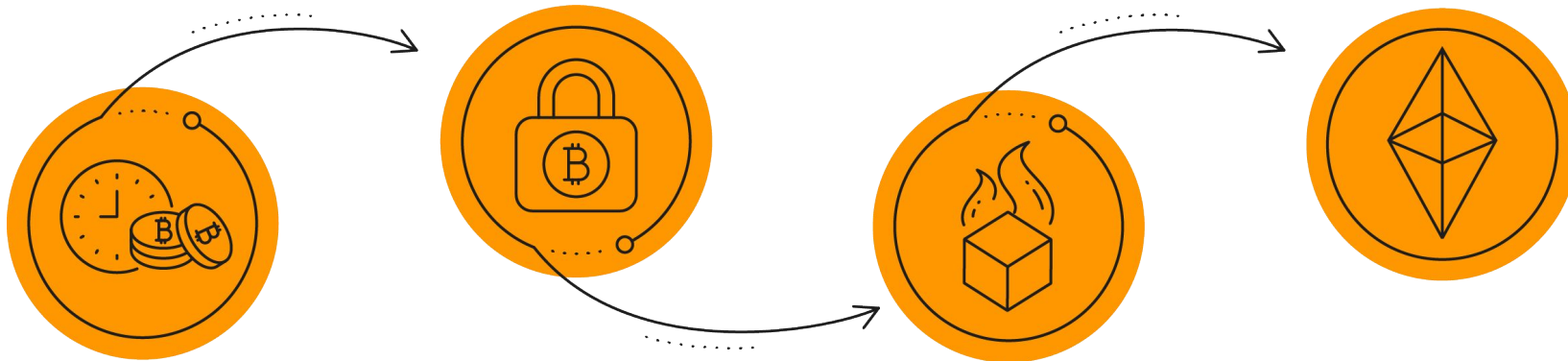
The act of converting computing power that costs real-world energy and money into a block with transactions in it.

The block is then submitted to the network for confirmation, and the block with the most work put into it gets added.

This is a very secure algorithm, but it's the most expensive in terms of resources. This is its biggest criticism.

# Consensus Algorithms

## Proof of Stake (PoS)



Very similar to PoW, only instead of contributing computational power, you “stake” some of the cryptocurrency for a period of time. Once past a minimum staking interval, you can then submit blocks to the rest of the network for confirmation.

“Staking” your coins means to lock them in a transaction that proves to the rest of the network that you are willing to “put your money where your mouth is” in order to be trusted to make blocks.

The biggest criticism is the “nothing at stake” problem, where block producers have nothing to lose for producing alternative versions or histories of the blockchain. Some versions of this algorithm include punishing cheaters by burning their stakes and not letting them get it back.

Despite this concern, much of the blockchain community is moving toward variations of PoS, including Ethereum.



# Instructor Demonstration

## Consensus Algorithms



## **Activity:** Turn and Teach Consensus Algorithms

In this activity, you will turn and teach the three consensus algorithms just covered.

**Suggested Time:**  
10 Minutes





**Time's Up!** Let's Review.

# Activity Review: Consensus Algorithms

What is the biggest strength of

01

**Proof of Work**

02

**Proof of Stake**

03

**Proof of Authority**





# Activity Review: Consensus Algorithms

What is the biggest weakness of

01

**Proof of Work**

02

**Proof of Stake**

03

**Proof of Authority**





# Instructor Demonstration

## Creating a Genesis Block

# Creating a Genesis Block

---

Today we are going to build our Ethereum blockchain.

We will start building  
the first block of  
the chain known as  
**Genesis Block.**




# Introducing Go Ethereum



# Creating a Genesis Block

The Go Ethereum tool is one of the three original implementations of the Ethereum protocol.

[Go Ethereum](#) [Install](#) [Downloads](#) [Documentation](#)



## What is Go Ethereum?

Go Ethereum is one of the three original implementations (along with C++ and Python) of the Ethereum protocol. It is written in Go, fully open source and licensed under the GNU LGPL v3.

See [our repository](#) and [downloads section](#) for the code!

## How can I use it?

Go Ethereum is available either as a standalone client called Geth that you can install on pretty much any operating system, or as a library that you can embed in your Go, Android or iOS projects.

See our [installation guide](#) or our [wiki pages](#) for details!

# Creating a Genesis Block

We will use the Go Ethereum tool via the `geth` command-line tool.

**geth** is the official Ethereum node software used to initialize, run and manage Ethereum nodes.

```
geth --dev console (geth)
0.00B gctime=0s livenodes=1 liveness=0.00B
INFO [11-15|15:02:36.429] Initialised chain configuration      config="{ChainID: 1337 Homestead: 0 DAO: <nil> DAOSup
port: false EIP150: 0 EIP155: 0 EIP158: 0 Byzantium: 0 Constantinople: 0 Petersburg: 0 Istanbul: 0 Engine: clique}"
INFO [11-15|15:02:36.430] Initialising Ethereum protocol      versions=[63] network=1337 dbversion=<nil>
WARN [11-15|15:02:36.430] Upgrade blockchain database version  from=<nil> to=7
INFO [11-15|15:02:36.443] Loaded most recent local header      number=0 hash=a890d2..d12429 td=1 age=50y7mo5d
INFO [11-15|15:02:36.443] Loaded most recent local full block  number=0 hash=a890d2..d12429 td=1 age=50y7mo5d
INFO [11-15|15:02:36.443] Loaded most recent local fast block  number=0 hash=a890d2..d12429 td=1 age=50y7mo5d
INFO [11-15|15:02:36.457] Allocated fast sync bloom           size=512.00MiB
INFO [11-15|15:02:36.458] Initialized fast sync bloom          items=11 errorrate=0.000 elapsed=96.739µs
INFO [11-15|15:02:36.458] Stored checkpoint snapshot to disk   number=0 hash=a890d2..d12429
INFO [11-15|15:02:36.459] started whisper v.6.0               seq=1 id=d71975f50276fb6c ip=127.0.0.1 udp=0 tcp=4956
INFO [11-15|15:02:36.459] New local node record                6
INFO [11-15|15:02:36.459] Started P2P networking               self="enode://0793af70a8273dce227aa5ef856b1eb22bbf6f1
170adf9e1c70766e7adf863024c21277714ecc4ff93f3c98d84addc32369cf9a9d2b76290b9927b8f1a4eb331@127.0.0.1:4956?discport=0"
INFO [11-15|15:02:36.460] IPC endpoint opened                  url=/var/folders/sr/y7j5gqms3s7cwwqhm8lgtgc0000gn/T/
geth.ipc
INFO [11-15|15:02:36.460] Transaction pool price threshold updated price=1000000000
INFO [11-15|15:02:36.460] Transaction pool price threshold updated price=1
INFO [11-15|15:02:36.460] Etherbase automatically configured  address=0x07f6746Ce7eDd2fDA8bB50428E4EB20EB4cb8b94
INFO [11-15|15:02:36.460] Commit new mining work               number=1 sealhash=27ba1a..78ceb0 uncles=0 txs=0 gas=0
fees=0 elapsed=81.739µs
INFO [11-15|15:02:36.460] Sealing paused, waiting for transactions
Welcome to the Geth JavaScript console!

instance: Geth/v1.9.6-stable/darwin-amd64/go1.13.1
coinbase: 0x07f6746ce7edd2fda8bb50428e4eb20eb4cb8b94
at block: 0 (Wed, 31 Dec 1969 19:00:00 EST)
datadir:
modules: admin:1.0 clique:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 shh:1.0 txpool:1.0 web3:1.0

> web3.fromWei(eth.getBalance(eth.coinbase))
1.15792089237316195423570985008687907853269984665640564039457584007913129639927e+59
>
```

# Creating a Genesis Block

---

The Go Ethereum tool is written in the Go programming language, fully open-source and licensed under the GNU LGPL v3.



**Don't worry, you don't need to learn Go!**  
You just have to know that it's super fast  
and has a cute mascot called Gopher.



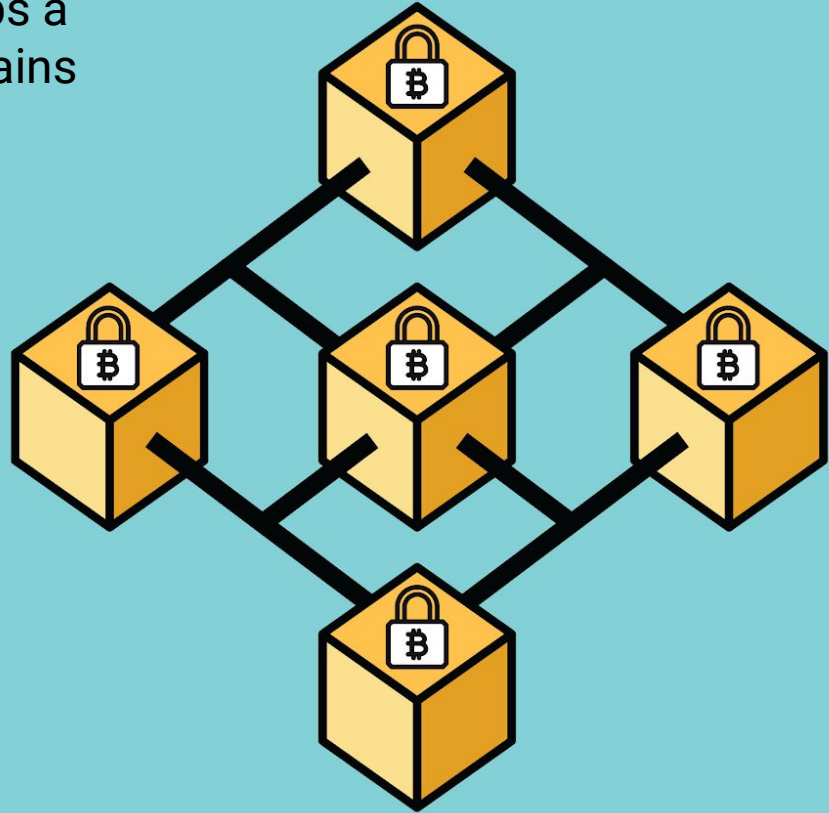


**Do You Remember What a Node is?**



# Creating a Genesis Block

A participant of the network that keeps a full copy of the blockchain and maintains the consensus rules of the network.





## **Activity:** Creating a Genesis Block

In this activity, you will create your own genesis configuration.

**Suggested Time:**  
10 Minutes

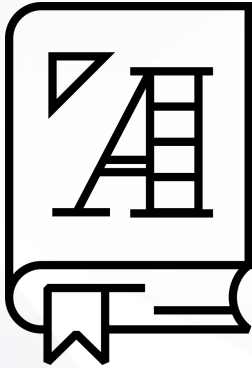




**Time's Up!** Let's Review.



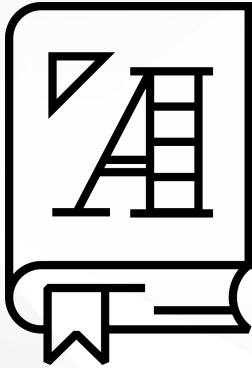
# **What is Important About the Genesis Block?**



It contains the **initial rules** for the blockchain network, like consensus algorithm, prefunded accounts, etc.



**What is the Point of Prefunding  
Accounts in the Genesis Block?**



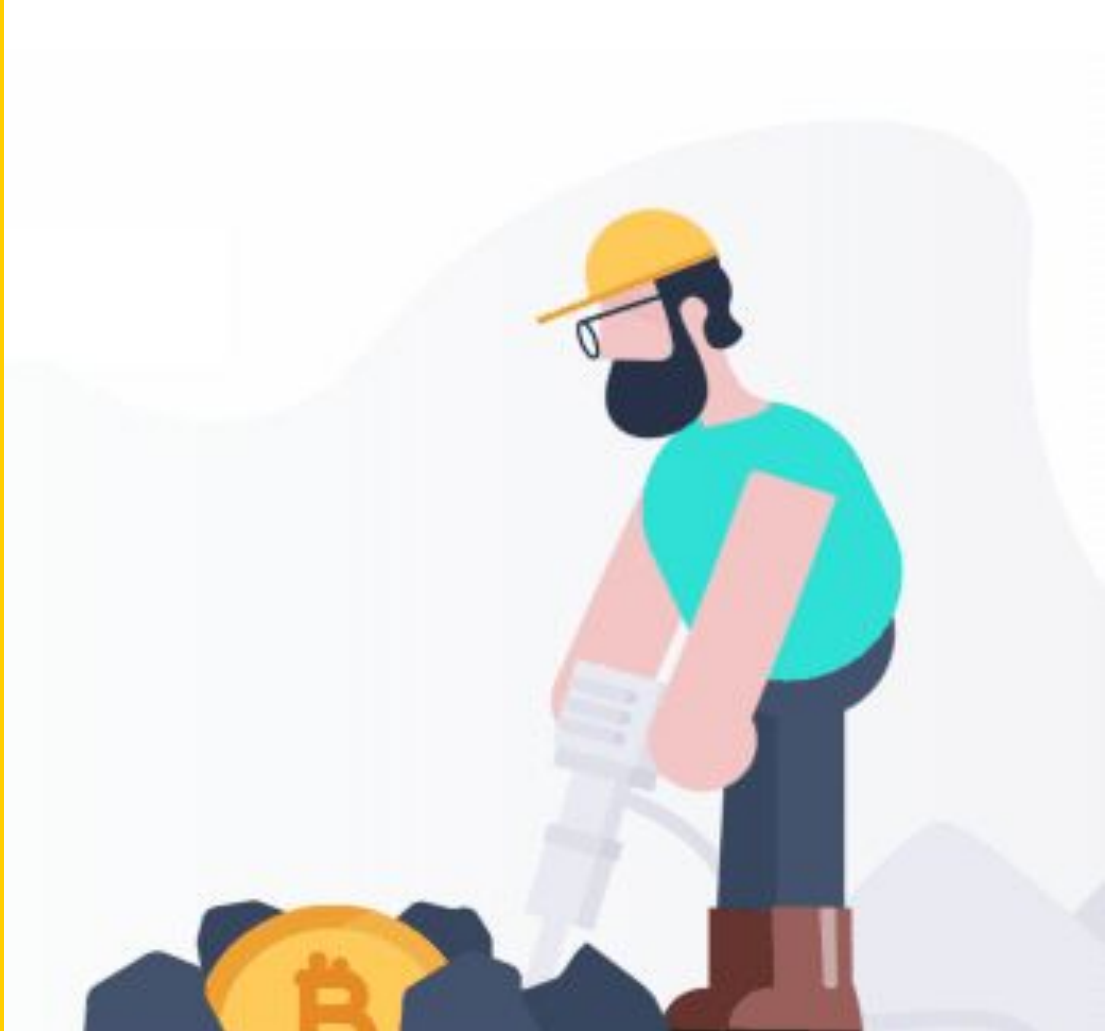
So that we have some **cryptocurrency** to test with right away; otherwise, we'll have to mine it manually (time-consuming).



**Since We Chose Proof of Work,  
What Mechanism are We Using to  
Create New Blocks?**



# Mining





# Instructor Demonstration

## Creating Two Nodes with Accounts



## **Activity:** Creating Two Nodes with Accounts

In this activity, you will create your own nodes and accounts for your custom blockchain network.

**Suggested Time:**  
15 Minutes

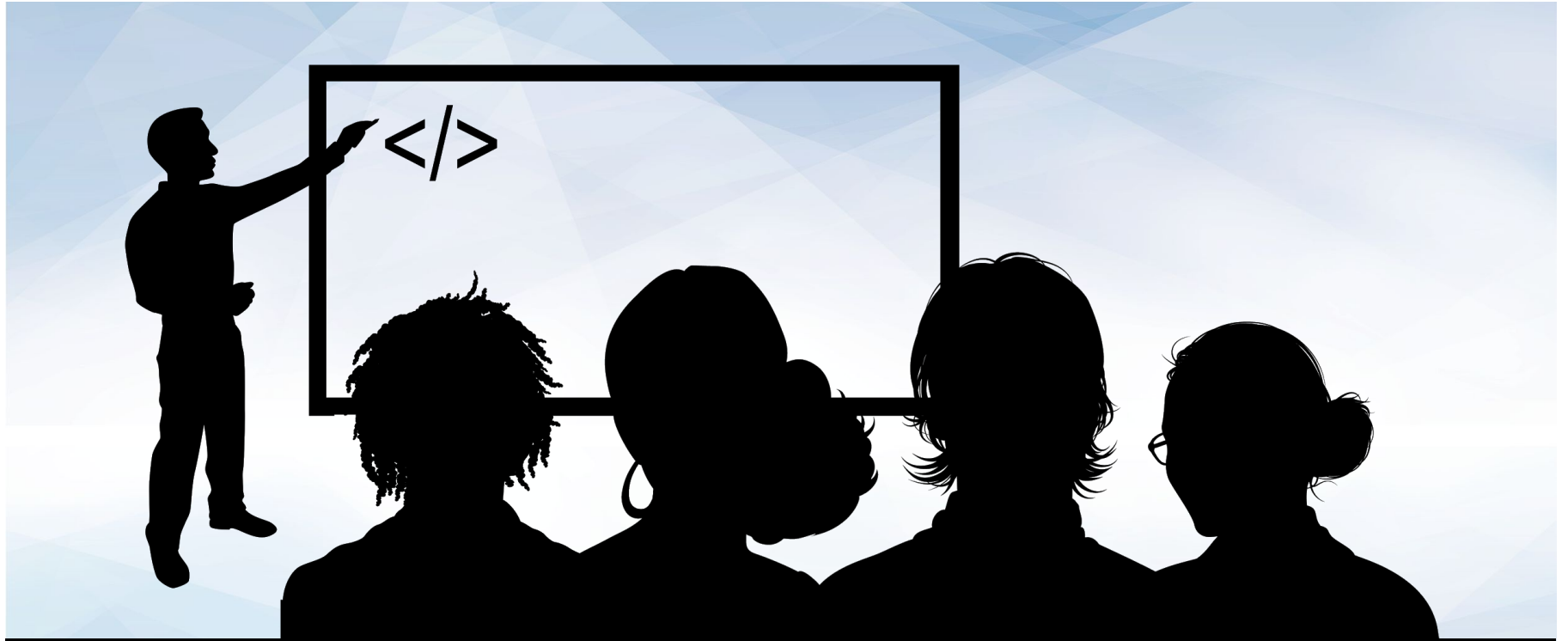




**Time's Up!** Let's Review.



Break



# Instructor Demonstration

## Starting the Blockchain



## **Activity:** Bringing the Blockchain to Life

In this activity, you will launch your own chains using the same techniques presented in the demo.

**Suggested Time:**  
15 Minutes





**Time's Up!** Let's Review.





# Instructor Demonstration

## Transacting on the Chain



## **Activity:** Transacting on the Chain

In this activity, you will connect MyCrypto to your chain and send a transaction!

**Suggested Time:**  
15 Minutes





**Time's Up!** Let's Review.



# Instructor Demonstration

## Recap



Questions?

*The  
End*